

DTIC FILE COPY

2

AD-A225 334

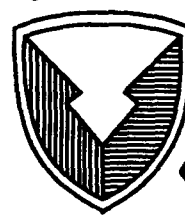
HDL-TR-2178

April 1990

Signal-Processing Techniques for Simulated Wideband Monopulse Radar Data

by Ronald J. Wellman
Jerry L. Silvius

DTIC
UNCLASSIFIED
AUG 17 1990
E



U.S. Army Laboratory Command
Harry Diamond Laboratories
Adelphi, MD 20783-1197

Approved for public release; distribution unlimited.

90

5

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) HDL-TR-2178			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Harry Diamond Laboratories		6b. OFFICE SYMBOL (if applicable) SLCHD-ST-SP	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State, and ZIP Code) 2800 Powder Mill Road Adelphi, MD 20783-1197			7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Army Materiel Command		7b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-0001			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO. P612120.H1600	PROJECT NO. AH16	TASK NO.
					WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) Signal-Processing Techniques for Simulated Wideband Monopulse Radar Data					
12. PERSONAL AUTHOR(S) Ronald J. Wellman and Jerry L. Silvious					
13a. TYPE OF REPORT Summary		13b. TIME COVERED FROM Jan 88 TO Jan 89		14. DATE OF REPORT (Year, Month, Day) April 1990	
15. PAGE COUNT 43					
16. SUPPLEMENTARY NOTATION AMS code: 612120.H160011 HDL project No.: 1E2913					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	High resolution, monopulse, MMW radar, signal processing		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) A computer simulation is presented of the signals received by a coherent, frequency-agile, two-coordinate monopulse radar from an array of point reflectors. A Fortran program for analyzing these simulated signals is used to obtain down-range and cross-range locations of the reflectors. Inverse Synthetic Aperture Radar (ISAR) techniques also are used to obtain a radar-cross-section-density image of a two-dimensional array of reflectors. The effects of system phase and amplitude noise also are examined.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Ronald J. Wellman			22b. TELEPHONE (Include Area Code) (202) 394-3130		22c. OFFICE SYMBOL SLCHD-ST-SP

DD Form 1473, JUN 86

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Contents

	Page
1. Introduction	5
2. Monopulse Radar Signals	5
2.1 <i>Monopulse Signals from Reflector Arrays</i>	5
2.2 <i>Data Simulation Program</i>	7
3. Data Processing Techniques	10
3.1 <i>Fourier Transform Processing</i>	10
3.2 <i>Monopulse Signal Processing</i>	13
3.3 <i>ISAR Processing</i>	14
4. Data Simulation Results	16
4.1 <i>High-Range-Resolution Results</i>	16
4.2 <i>Monopulse Results</i>	21
4.3 <i>ISAR Results</i>	24
5. Conclusions	27
References	28
Appendix A. Listing of stepped-frequency monopulse signal calculation program	29
Distribution	41

Figures

1. Monopulse calculation beam geometry	6
2. Range imaging using 64-point FFT with a rectangular window and imaging with padded 128-point FFT	12
3. Imaging with padded 128-point FFT using Hamming window	13
4. Monopulse sum and normalized azimuth difference pattern	14
5. Baseline single-point reflector image using 128-point FFT with Hamming window for noise comparison	17
6. Effect of -30-dB pulse-to-pulse amplitude noise below signal on 128-point FFT single-point reflector range image	17
7. Effect of -20-dB pulse-to-pulse amplitude noise below signal on 128-point FFT single-point reflector range image	18
8. Effect of -10-dB pulse-to-pulse amplitude noise below signal on 128-point FFT single-point reflector range image	18

9. Effect of 40-dB system signal-to-noise ratio on 128-point FFT single-point reflector range image	19
10. Effect of 30-dB system signal-to-noise ratio on 128-point FFT single-point reflector range image	19
11. Effect of 20-dB system signal-to-noise ratio on 128-point FFT single-point reflector range image	19
12. Effect of 10-dB system signal-to-noise ratio on 128-point FFT single-point reflector range image	19
13. Effect of 5° pulse-to-pulse phase variation on 128-point FFT single-point reflector range image	20
14. Effect of 10° pulse-to-pulse phase variation on 128-point FFT single-point reflector range image	20
15. Effect of a 20° pulse-to-pulse phase variation on 128-point FFT single-point reflector range image	20
16. Effect of a combination of 20-dB system signal-to-noise ratio, -20-dB pulse-to-pulse amplitude noise and 5° pulse-to-pulse phase variation on 128-point FFT single-point range image	21
17. Effect of 40-dB system signal-to-noise ratio on determining off-axis target position using monopulse calibration on high-range-resolution target image data	22
18. Effect of 30-dB system signal-to-noise ratio on determining off-axis target position using monopulse calibration on high-range-resolution target image data	23
19. Effect of 20-dB system signal-to-noise ratio on determining off-axis target position using monopulse calibration on high-range-resolution target image data	23
20. Three-dimensional ISAR image of target array for noiseless case	25
21. Three-dimensional ISAR image of target array for 20-dB input signal-to-noise ratio	26

Tables

1. Selected Data Window Types Showing Performance Specification	13
2. Effect of Different Types of Noise on High-Range-Resolution FFT Imaging	21
3. Comparison of Input Target Location with the Amplitude and Position Derived from ISAR Image	25
4. Comparison of Average Output Noise Level for ISAR Processing with Input Noise Level	26

1. Introduction

Research is being conducted at the Harry Diamond Laboratories (HDL) on millimeter-wave radar target signatures. A coherent, frequency-agile, two coordinate, amplitude comparison 95-GHz monopulse radar was designed and built at HDL for this purpose. To simulate the data that will be obtained with this radar, a computer program was written that calculates the signal return from a complex array of point reflectors. This program can simulate frequency agility, both AM and FM system noise, and signature data generated with a rotating target array. Section 2 of this report describes the data simulation program.

Two additional programs were written to analyze the data generated by the simulation program. These programs form the basis of some of the analysis programs that will be required for processing data obtained with the radar. The first program uses a Fast Fourier Transform to obtain the down-range location of the simulated target array, and the sum and difference patterns generated by the monopulse radar to obtain the cross-range location. The second program uses frequency-stepped data generated as a target array is rotated on a turntable to generate a two-dimensional radar-cross-section-density image of the target array. This is done with Inverse Synthetic Aperture Radar (ISAR) techniques. These programs are described in section 3 of this report.

In section 4 of this report we discuss the results of using the programs of section 3 with the data generated by the program in section 2. The discussion includes high-range-resolution, monopulse, and ISAR results.

2. Monopulse Radar Signals

2.1 Monopulse Signals From Reflector Arrays

Consider a four-beam two-coordinate amplitude comparison monopulse radar illuminating an array of M point reflectors. The radar is assumed to be located at the origin of a coordinate system and the j^{th} reflector is located at position $P_j(x_j, y_j, z_j)$ (see fig. 1). If the phase of the transmitted signal is θ_o and that of the received signal is θ , then the shift in phase of a signal propagating from the radar to the reflector and back is given by

$$\Delta\theta = \theta - \theta_o \quad (1)$$

and equals

$$\Delta\theta = \left(\frac{2\pi f_k}{c}\right)(2|P_j|) = \left(\frac{4\pi f_k}{c}\right)[x_j^2 + y_j^2 + z_j^2]^{1/2} \quad (2)$$

where c is the speed of light and f_k is the frequency.

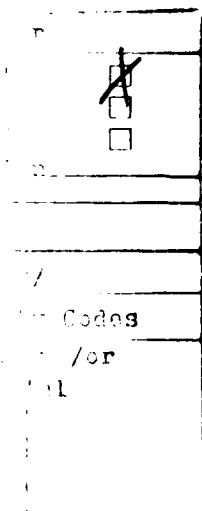
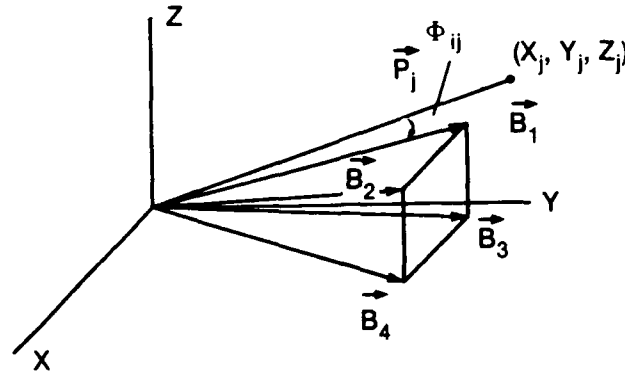


Figure 1. Monopulse calculation beam geometry.



The amplitude of the received signal is dependent on the distance, $|P_j|$, of the reflector from the origin, the reflectivity, ρ_j , of the reflector, and on its position in the radar beam. We assume that each beam of the monopulse radar has circular symmetry about the individual beam axes and that in the far field of the antenna the normalized field intensity at an angle, Φ , from this axis is the known antenna pattern, $E(\Phi)$. If B_i is a vector with the same magnitude as P_j pointed along the axis of the i^{th} beam of a monopulse radar, then the angle between the i^{th} beam axis and the j^{th} reflector is given by

$$\cos(\Phi_{ij}) = \frac{P_j \cdot B_i}{|P_j|^2} \quad (3)$$

The field intensity for a point P_j is then calculated from the angle Φ_{ij} for each beam.

In figure 1, the monopulse axis is taken to be along the y-axis and the individual beams, B_i 's, are numbered 1 to 4. The squint angle, α_i , is the angle between each B_i and the x-y or the y-z plane.

The electric field intensity at the receiver for the monopulse sum is therefore

$$\begin{aligned} X_{k\Sigma}(f_k) &= \sum_{i=1}^4 \sum_{j=1}^M \frac{\rho_j E(\Phi_{ij})^2}{|P_j|^2} \exp\left(\frac{i4\pi f_k |P_j|}{c}\right) \\ &= \sum_{i=1}^4 \sum_{j=1}^M \frac{\rho_j E(\Phi_{ij})^2}{|P_j|^2} \left[\cos\left(\frac{4\pi f_k |P_j|}{c}\right) \right. \\ &\quad \left. + i \sin\left(\frac{4\pi f_k |P_j|}{c}\right) \right] = I_{k\Sigma} + iQ_{k\Sigma} \quad (4) \end{aligned}$$

If the initial phase, θ_o , of the transmitted beam is chosen to be zero, $I_{k\Sigma}$ is the in-phase and $Q_{k\Sigma}$ is the quadrature-phase portion of the received sum field intensity. Similarly, the azimuth and elevation differences, $X_{k\Delta Z}$ and $X_{k\Delta E}$, are given by

$$X_{k\Delta AZ}(f_k) = \sum_{j=1}^M \left[E(\Phi_{1j})^2 + E(\Phi_{3j})^2 - E(\Phi_{2j})^2 - E(\Phi_{4j})^2 \right] \left[\frac{\rho_j}{|P_j|^2} \cos\left(\frac{4\pi f_k}{c} |P_j|\right) + i \frac{\rho_j}{|P_j|^2} \sin\left(\frac{4\pi f_k}{c} |P_j|\right) \right] \quad (5)$$

$$= I_{k\Delta AZ} + iQ_{k\Delta AZ} .$$

and

$$X_{k\Delta EL}(f_k) = \sum_{j=1}^M \left[E(\Phi_{1j})^2 + E(\Phi_{2j})^2 - E(\Phi_{3j})^2 - E(\Phi_{4j})^2 \right] \left[\frac{\rho_j}{|P_j|^2} \cos\left(\frac{4\pi f_k}{c} |P_j|\right) + i \frac{\rho_j}{|P_j|^2} \sin\left(\frac{4\pi f_k}{c} |P_j|\right) \right] \quad (6)$$

$$= I_{k\Delta EL} + iQ_{k\Delta EL} .$$

Equations (4), (5), and (6) form the basis for the monopulse signal calculations.

The normalized electric field pattern of each beam, $E(F)$, is taken to be that for a cosine aperture distribution [1]:

$$E(\Phi) = \frac{\pi}{4} \left[\frac{\sin(\psi + \pi/2)}{\psi + \pi/2} + \frac{\sin(\psi - \pi/2)}{\psi - \pi/2} \right] , \quad (7)$$

where $\psi = (a/\lambda) \sin \Phi$, a is the aperture diameter, and λ is the wavelength. The wavelength used for the calculation was 3.16 mm and the aperture diameter was 15.75 cm. The squint angle, α_s , was set at 7.44 mrad, slightly less than the value for a maximum monopulse error slope [1].

2.2 Data Simulation Program

The data simulation program uses the equations generated in the previous section to calculate the monopulse radar signal returned from an array of point reflectors. The main program is composed of two sections. The first calculates the signal return for a frequency-agile monopulse radar illuminating a stationary target array. The second extends this calculation to that for a rotating target array. Both sections also allow for the addition of amplitude or phase noise to the ideal signal calculations to simulate radar data with a finite signal-to-noise ratio.

The stepped-frequency monopulse signal calculations are made using the program listed in appendix A. The program requires the user to input the positions of each point of the target array (geometric parameters) and the reflectivity of each point (target parameters). The user also enters the number of frequency steps, N , and the frequency step size. The program calculates the summations given by equations (4), (5), and (6) for N frequencies beginning at 94.44 GHz. The values calculated for the in-phase and quadrature components, I and Q , for each frequency are placed in a complex array which is then stored in a disc file. The data in this file are used in separate programs for analyses that will be discussed in the following sections of this report.

The second section of the simulation program generates monopulse radar signals for an array of point isotropic reflectors on a turntable. In this case, the target array and the radar line-of-sight are assumed to lie in the plane of the turntable. The radar line-of-sight also is taken to pass through the axis of rotation, which is perpendicular to the plane of the turntable. With the data generated by this program, ISAR techniques can be used to calculate high-resolution images of the target in both down-range and cross-range configurations.

The program starts the calculation by recording the amplitude and phase history of a point reflector as it moves on a circle about the center of rotation. The value of $|P_j|$ in equation (4) is computed in the following manner. The range from the radar to the center of rotation of the target array is defined by the vector y_o , the radial distance of the j^{th} point to the center of rotation is taken to be r_j , and the initial angle of each point reflector is w_{oj} . The angle w_{oj} is the angle between r_j and a line perpendicular to y_o and passing through the center of rotation. The simulation program computes $|P_j|$ for each point reflector and from it the radar signal (eq (4), (5), and (6)) for N frequency steps. Each w_{oj} is then incremented by an angle, Δw , where Δw is defined to be negative for clockwise rotation, and the calculation is repeated. The resulting P_j following m angular increments is

$$|P_j| = [r_j^2 + y_o^2 + 2 r_j y_o \sin(w_{oj} + m\Delta w)]^{1/2} . \quad (8)$$

The calculation is then repeated for as many aspect angles as is desired to generate the ISAR image. The simulation consisted of 64 sets of different aspect angles separated by Δw with 64 frequency steps of 10 MHz for each aspect angle. The calculated I , Q values for each frequency and each position were stored into a two-dimensional complex array. This array was then stored in a file to be processed by another program, which will be discussed in section 3.3.

Once the basic signal calculations are made, another section of the program allows the addition of different types of noise to the calculated monopulse

sum and difference signals to simulate actual measurement conditions. For a frequency $f_k = f$, the equations for the sum and difference I , Q values (eq (4), (5), and (6)) were modified and expressed in the form shown below to add various types of noise components:

$$I_{\Sigma} = (A + A') \cos(\theta + \theta') + D' \cos \theta_3 \quad (9)$$

$$Q_{\Sigma} = (A + A') \sin(\theta + \theta') + D' \sin \theta_3 \quad (10)$$

$$I_{\Delta Z} = B \cos \theta_1 + B' \cos \theta_4 \quad (11)$$

$$Q_{\Delta Z} = B \sin \theta_1 + B' \sin \theta_4 \quad (12)$$

$$I_{\Delta EL} = C \cos \theta_2 + C' \cos \theta_5 \quad (13)$$

$$Q_{\Delta EL} = C \sin \theta_2 + C' \sin \theta_5 \quad (14)$$

where A , B , C , θ , θ_1 , and θ_2 are the sum, azimuth difference, and elevation difference signal amplitudes and phases, respectively. Pulse-to-pulse amplitude and phase noise are represented by the Gaussian-distributed variables A' and θ' , which have been added to only the sum channel signals in our simulation. For the difference channel signals, the dominant noise is system noise, which is set by the system noise figure. This is so because the difference channel signals are normally small for on-axis targets, producing a much smaller signal-to-noise ratio in the difference channels. In addition, both pulse-to-pulse amplitude and phase noise will affect all the channels equally, so that the normalized difference signal, which determines off-axis distance, will not be affected by the addition of pulse-to-pulse noise. Noise contributions due to poor system signal-to-noise ratio and random phase noise are represented by the random noise amplitude components D' , B' , and C' and the random phase components θ_3 , θ_4 , and θ_5 .

The different types of noise sources were compared by running simulations with each noise separately and in one case with all types of noise included. Simulations were run with AM noise for pulse-to-pulse signal-to-noise ratios of 30, 20, and 10 dB. System signal-to-noise ratios of 40, 30, 20, and 10 dB were used to analyze the effect of poor signal-to-noise ratio on system performance. In addition, pulse-to-pulse phase errors of 5, 10, and 20 deg were added to the signal in another set of simulations. The final simulation was run using a system signal-to-noise ratio of 20 dB, a pulse-to-pulse signal-to-AM-noise ratio of 20 dB and a pulse-to-pulse phase error of 5°.

The data generated in these simulations are used in data processing programs that generate high-range-resolution images and two-dimensional radar ISAR images of the target arrays. In addition the data are used in calculating the off-axis monopulse response for the simulated radar. The results of these simulations will be discussed in section 4 of this report.

3. Data Processing Techniques

3.1 Fourier Transform Processing

The basic techniques for processing the frequency-stepped monopulse radar data rely on the use of the Fast Fourier Transform (FFT) algorithm to generate high resolution range gates. For evaluating turntable data, a two-dimensional FFT can be used to generate high resolution two-dimensional ISAR images of the target. The applications of both of these processing techniques to the simulated radar data will be discussed in this section.

The FFT is a useful tool for performing both time-domain and frequency domain signal analysis. The most common use of the FFT is to transform time domain signals into their corresponding spectral components. This type of transform generates a function with amplitude and phase values over a defined frequency range. The basic Fourier transform pair equations are given by the following integrals [2]:

$$x(t) = \int_{-\infty}^{\infty} X(f) \exp(j2\pi ft) df \quad (15)$$

$$X(f) = \int_{-\infty}^{\infty} x(t) \exp(-j2\pi ft) dt, \quad (16)$$

where the first integral defines the inverse Fourier transform and the second integral the direct Fourier transform. For sampled data, the discrete Fourier transform (DFT) is used. These transforms are defined by the following summations [2]:

$$x(n\Delta t) = \Delta f \sum_{k=0}^{N-1} X_d(k\Delta f) \exp(j2\pi k\Delta f n\Delta t), \text{ and} \quad (17)$$

$$X_d(k\Delta f) = \Delta t \sum_{n=0}^{N-1} x(n\Delta t) \exp(-j2\pi k\Delta f n\Delta t), \quad (18)$$

where

- N = number of samples,
- Δt = time between samples,
- Δf = sample interval in the frequency domain,
- n = the time sample index $n = 0, 1, 2, \dots, N-1$,
- k = the index for discrete frequency components, $k = 0, 1, 2, \dots, N-1$,
- $x(n\Delta t)$ = discrete set of time samples that defines the waveform,
- $X(k\Delta f)$ = the set of Fourier coefficients obtained by the DFT of $x(n\Delta t)$.

These equations form the basis for computer calculations of the Fourier transform. Many computers have a built-in FFT algorithm that is much faster than a direct calculation of the summations. This analysis is not intended to be a dissertation on the Fourier transform, but we will use the properties of the Fourier transform. These properties are discussed in greater detail elsewhere [2].

Since the Fourier transform has an inverse transform, it is apparent that one can transform from the frequency domain to the time domain as well as from the time domain to the frequency domain. With a frequency-agile coherent system, the data represent information in the frequency domain, and when the FFT is used, the data are transformed into the time domain. Using this technique, high-resolution range gates are generated from wide-bandwidth stepped-frequency signals.

In transforming the data from the frequency domain to the time domain, a rectangular function in the frequency domain is transformed into a $\sin x/x$ distribution in the time domain in a similar way as a rectangular pulse in the time domain transforms into a $\sin x/x$ distribution in the frequency domain. To demonstrate this effect, a computer simulation was run to calculate the signal return for two targets located at 93.685 and 101.297 m, each having a radar cross section of 1 m². The simulation calculated the signal return for 64 frequency steps of 10 MHz beginning at 94.44 GHz. The inverse DFT was used to generate the data shown in figure 2(a). The plot gives the calculated amplitude of each scatterer as a function of the point number of the DFT.

The resolution cell width is given by [3]

$$\Delta R = c/2N\Delta f , \quad (19)$$

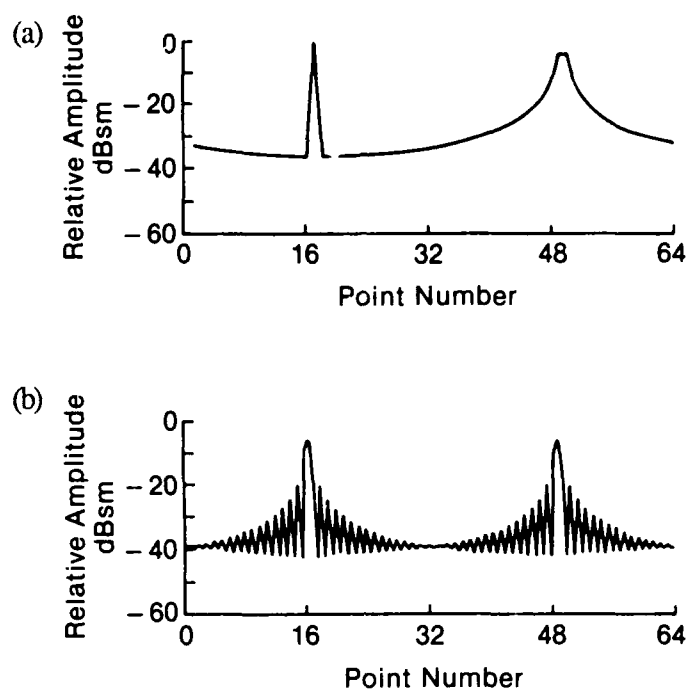
where c is the speed of light, N is the number of frequency steps, and Δf is the frequency step size. For 64 steps of 10 MHz the resolution cell width or range resolution is 0.234 m. The total range coverage of the DFT is given by [3]

$$R = c/2\Delta f . \quad (20)$$

For the 10-MHz step size, the total range coverage is 14.99 m. This means that the target range is broken up into range intervals approximately 15 m in extent that overlap every 15 m; that is, a target at 10 m will image at the same position as targets at $15n + 10$ m, where n is an integer. Therefore, this technique does not give absolute range, but only relative range within the 14.99 m of coverage.

It is apparent from fig. 2(a) that the two points are not being imaged in the same manner. This occurred because the first or closest point was placed in the center of a range bin, whereas the second point was placed between two range bins. For the point in the center of the bin, the peak amplitude occurs at the center of a range cell, and the samples near the peak occur at the nulls

Figure 2. Range imaging using 64-point FFT with a rectangular window and imaging with padded 128-point FFT.



of the $\sin x/x$ distribution. For the other point the amplitude is reduced by about 4 dB, and the samples around the peak now occur at the maxima of the side lobes of the $\sin x/x$ distribution.

This amplitude variation is caused by the location of the target with respect to the range gates generated by the FFT. For a target located at the center of a gate, the amplitude response is maximum. For a target located in between gates, the amplitude response is determined by both gates. To reduce the amplitude variation, overlapping range gates with gates lying in between the original gates need to be generated. This is simply accomplished by extending the original record length with a set of samples that are identically zero. For our purposes, 32 zeros are added at the beginning and the end of the data set. This process is referred to as padding, and in this case a 128-point DFT is performed. The results of this procedure are shown in figure 2(b). The amplitude of both points is now the same, but for a point located between them, there would still be an amplitude reduction. There is also an overall reduction in amplitude of 6 dB. The peaks and nulls of the $\sin x/x$ distribution are now easily distinguishable. To further enhance the response and reduce the amplitude variation, more zeros could be added, but the computation time would be greatly increased. For all subsequent data analyses in this report, 64 zeros were added to the data before processing.

The images shown in figure 2(b) can be improved further by changing the window function used in processing the data. For the initial data processing, a rectangular window function was used, which generated sidelobes begin-

ning at 13 dB down from the peak. In generating range images of complex targets, this type of sidelobe performance would greatly restrict the usefulness of the images. There are a variety of other potential window functions; some of the most common are listed in table 1. The table includes the sidelobe performance, corresponding reduction in peak level, and 3-dB mainlobe width of the window functions where $B = 1/T$. For this application, the Hamming window provides the best sidelobe performance. Use of this window function is demonstrated in figure 3 for the same simulation points as in figure 2. The FFT point number for this figure and all subsequent figures has been replaced by relative range, since the point number is proportional to the range within the 15-m window. Easily discernible differences are a reduction in the first sidelobe level of 29 dB, a reduction in peak level of 5.35 dB, and an increase in the 3-dB width of the main response.

In the remainder of this report, the Hamming window will be used in conjunction with padding to improve the down-range imaging properties of the data analysis. The total reduction in peak response due to the windowing and padding is 11.35 dB, and the down-range images will reflect this reduction in amplitude.

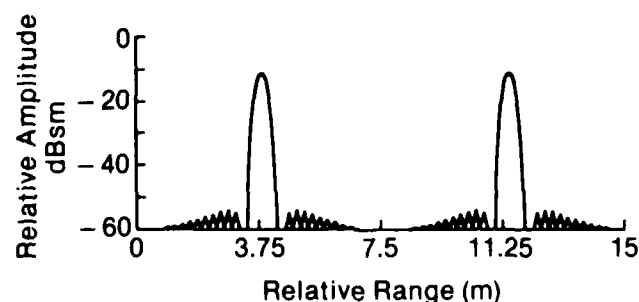
3.2 Monopulse Signal Processing

The ideal monopulse radar calculates the target off-axis angle error by a comparison between the difference channel signal and the sum channel amplitude at each range gate. When a complex target with range extent less than the range cell determined by the transmitted pulsewidth is illuminated with a frequency-agile monopulse radar, the target can be resolved in range with higher resolution by using the DFT algorithm. The absolute range is then determined by the time delay to the sampling window, and the relative range

Table 1. Selected Data Window Types Showing Performance Specification

Window type	Maximum sidelobe (dB)	Peak amplitude	Mainlobe 3-dB width
Rectangular	-13.2	T	0.86 B
Triangular	-26.7	0.5T	1.27 B
Cosine (Hanning)	-31.6	0.5T	1.39 B
Hamming	-41.9	0.54T	1.26 B

Figure 3. Imaging with padded 128-point FFT using Hamming window.



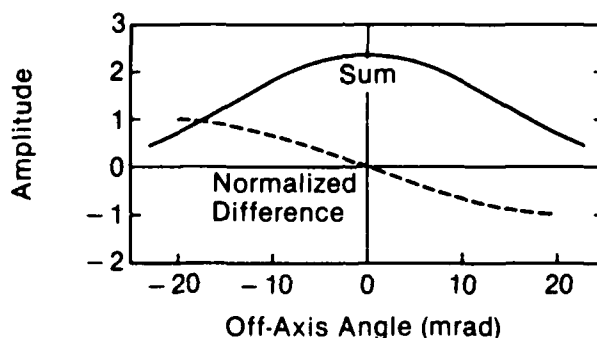
within that window is further resolved using the DFT. In addition, the difference-channel data can be processed using the DFT to provide a down-range profile of both azimuth and elevation error signals. By applying a cross-range calibration to these error signals, we can determine the cross-range position of the scattering center.

To generate a monopulse calibration curve for the off-axis angle error, a monopulse simulation was run using a single-point scatterer as the target. The target was positioned at a range of 100 m, at the center of the elevation axis, and translated through an azimuth angle of -20 to $+20$ mrad from the monopulse axis. The azimuth error signal was calculated along with the sum channel signal. An off-axis calibration curve was generated by normalization of the difference channel signal to the sum channel signal. The results of this calculation are shown in figure 4. Since the monopulse beam pattern was assumed to be symmetric in both elevation and azimuth, the normalized difference channel calibration curve can be used to calculate the angle error for both axes. The results of applying this technique to the analysis of a complex target are discussed in section 4.2 of this report.

3.3 ISAR Processing

The basic imaging technique used in generating a two-dimensional target image from coherent frequency-agile rotating target data is very similar to that employed in generating a synthetic aperture radar (SAR) image. In the latter method, the radar is moved around or past the target, and the return signal amplitude and phase are recorded as a function of radar position. The movement of the radar synthesizes a larger antenna aperture, which improves the radar resolution. In the ISAR method, the target is rotated and the radar is held fixed while the amplitude and phase history of the target are recorded. To generate the image, a two-dimensional FFT is used. The FFT is performed on the frequency-stepped data to provide a down-range amplitude and phase history of the individual scatterers as they are rotated through a fixed angle. Included in the phase history is the sampled Doppler frequency of each scatterer due to its relative motion with respect to the radar. The second FFT is used to analyze the Doppler frequencies, both positive and negative, that

Figure 4. Monopulse sum and normalized azimuth difference pattern.



exist in the phase history. These Doppler frequencies can be related directly to the off-axis distance of the individual scatterers. The details and applications of high resolution imaging techniques have been reported elsewhere [3–6]. We will use these results and discuss the application of these methods in analyzing the simulated and actual radar data.

To generate the ISAR image, a set of frequency-stepped data are generated for many different aspect angles. For our simulations, 64 different frequencies make up the frequency stepped data, and 64 different aspect angles are used for generating the cross-range resolution. The down-range resolution is determined by the frequency-stepped data as described earlier, whereas the cross-range resolution is determined by the total angle of observation. The cross-range resolution is given by [3]

$$\Delta x = c/2FM\Delta w , \quad (21)$$

where c is the speed of light, F is the average frequency during a frequency ramp, M is the number of different aspect angles observed, and Δw is the angular increment between successive frequency ramps. The total cross-range distance imaged is given by [3]

$$x = c/2F\Delta w . \quad (22)$$

There are some restrictions that must be adhered to during the generation of the data if the image quality is to be preserved. The rotation rate has an effect on the quality of both the down-range and cross-range images. If the rotation rate is too high, there will be significant down-range and cross-range image degradation, as discussed next.

The down-range image is generated from the phase history of the target as it moves during the frequency stepping. For a stationary target, the target is imaged accurately, but if there is target motion, the Doppler frequency can distort the down-range position. For accurate imaging, the Doppler shift from target motion must be kept below the effective sampling frequency of the frequency-stepped waveform. The Doppler shift from target motion is given by

$$F_d = 2r\omega/\lambda , \quad (23)$$

where r is the target radius, ω is the rotation rate in radians per second, and λ is the wavelength. The effective sampling frequency of the frequency-stepped waveform is given by

$$F_I = PRF/N , \quad (24)$$

where PRF is the pulse repetition rate and N is the number of steps required to generate a down-range image. To maintain a reasonable image quality, F_d

must be kept less than one-eighth of F_r . This means that the rotation rate is limited by

$$\omega \leq PRF\lambda/16\pi N \quad (25)$$

For a radar operating at a wavelength of 3.16 mm, using 64 frequency steps, and a repetition rate of 10 kHz, and with a maximum target radius of 5 m, the maximum rotation rate is equal to 0.0062 rad/s or one complete revolution of the target every 17 min. In the rotating target simulation reported here, the frequency-stepped data were generated while the target was stationary, so there is no apparent Doppler frequency included in the stepped frequency data. The target aspect angle was incremented by 0.00881° between ramps to generate the cross-range resolution. The Doppler is due to the motion between frequency ramps, and it is contained in the aspect angle data. The cross-range resolution is 0.16 m.

The ISAR data are processed in the following manner. The amplitude and phase are calculated for each frequency as the frequency is stepped, and the 64 data points are stored in a complex array. This procedure is repeated for each aspect angle, thereby generating a 64×64 complex array. This complex array is then windowed in both dimensions with a Hamming window being used to reduce sidelobes. The data are then padded with 32 zeros at the beginning and end of each frequency ramp, and further padded with 32 ramps of zeros at the beginning and end of the aspect angle data. This generates a complex array with 128×128 elements. A DFT algorithm is applied to the 128 sets of frequency-stepped data, and the resulting array is then transformed by aspect angle. The array must be inverted to order the Doppler frequency data, which correspond to the cross-range coordinate, to produce the proper image. The resulting amplitude data are corrected by adding 22.8 dB to compensate for the reduction in signal amplitude due to windowing and padding. The array is plotted in down-range and cross-range distance to give a two-dimensional radar-cross-section image of the target. The results of this simulation will be discussed in the next section.

4. Data Simulation Results

4.1 High-Range-Resolution Results

The range imaging capability of FFT processing was shown in figure 3. This section will discuss the effects of different types of noise sources on the quality of images that can be generated with FFT processing. A baseline simulation was run with a point reflector (with radar cross section of 1 m^2) placed 97.433 m from the radar. The data were generated as discussed in

section 3.1. Figure 5 shows the results of the simulation. The amplitude of the peak is reduced by 11.35 dB because of windowing and padding, with a range response indicative of the Hamming window function. The calculations were performed with double precision arithmetic to minimize processing errors due to truncation, and there were no noise components added to the signal. This figure will be the baseline for comparing the effects of the different types of noise to the ideal case.

The first type of noise considered is that from pulse-to-pulse amplitude fluctuations, which can be considered to be pure AM noise. The results of adding this kind of noise to the calculated data are shown in figures 6, 7, and 8. The AM signal-to-noise ratios are 30, 20, and 10 dB, respectively; this ratio was varied by changing the noise contribution while keeping the signal level constant. The figures indicate an increase in the baseline noise level as the signal-to-noise ratio is changed from 30 to 20 to 10 dB. This increase is proportional to the change in signal-to-noise ratio, but there is no apparent change in the peak amplitude response. For the 10-dB fluctuation case, the amplitude of the peak response still exceeds the average noise level by about 28 dB. This demonstrates the signal-to-noise ratio improvement that can be generated by coherently processing 64 different samples of the same signal plus Gaussian distributed noise. Figure 6 shows that a fluctuation ratio of 30 dB does not produce significant degradation from the noiseless case, since the noise level is almost equivalent to the maximum sidelobe level generated with the Hamming window.

Figure 5. Baseline single-point reflector image using 128-point FFT with Hamming window for noise comparison.

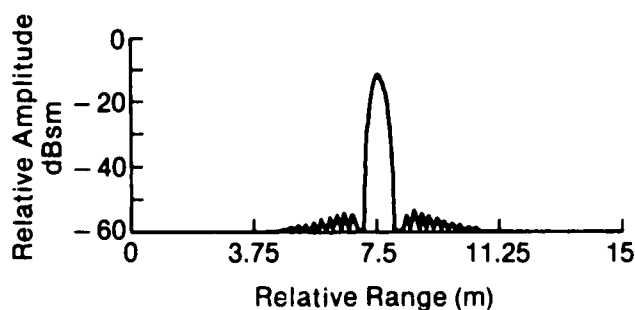


Figure 6. Effect of -30 dB pulse-to-pulse amplitude noise below signal on 128-point FFT single-point reflector range image.

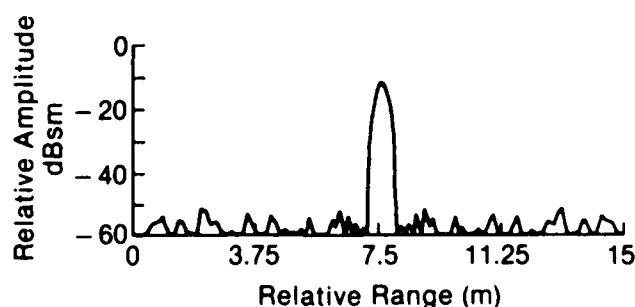


Figure 7. Effect of -20 dB pulse-to-pulse amplitude noise below signal on 128-point FFT single-point reflector range image.

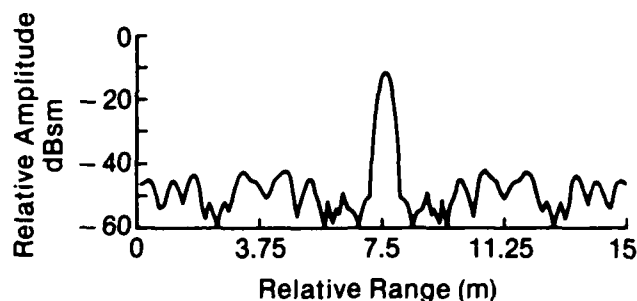
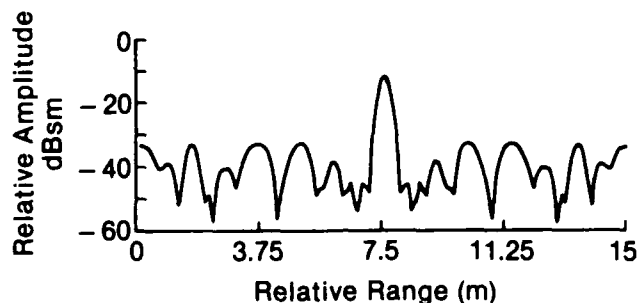


Figure 8. Effect of -10 dB pulse-to-pulse amplitude noise below signal on 128-point FFT single-point reflector range image.



Another interesting feature of these figures is the symmetry of the baseline noise around the peak response. For pure AM noise, the detected phase is not affected by the noise modulation. Since the detected phase determines the location of the target return, the noise will simulate a fluctuating target at the real target location. The resulting distribution will be symmetrical in time about the target location. Pure AM noise will be only one component of the total noise in a real radar system, but this simulation is useful in determining how much amplitude noise is acceptable.

The second type of noise to be modeled is system noise. As described earlier, system noise was simulated by adding a random amplitude fluctuation with a random phase component to both the I , Q -channels. System noise will affect all data channels randomly and will be the dominant noise factor for small signal levels. Sample simulations were run for signal-to-noise ratios of 40, 30, 20, and 10 dB. The results are shown in figures 9 through 12, respectively. These figures closely match those done for pulse-to-pulse amplitude noise, except that the noise distribution is no longer symmetrical about the point location. This effect is caused by the variation in the total signal phase created by adding both a random amplitude and random phase noise modulation. As with the pulse-to-pulse case, the noise does not cause much image degradation even at 10 dB, but there is a significant loss of dynamic range in the image when the noise level exceeds 20 dB.

The final type of noise which was modeled separately was that caused by random phase errors. These can be caused by loss of phase lock, by target or radar motion during data acquisition, or by phase errors in the I , Q detection

Figure 9. Effect of 40-dB system signal-to-noise ratio on 128-point FFT single-point reflector range image.

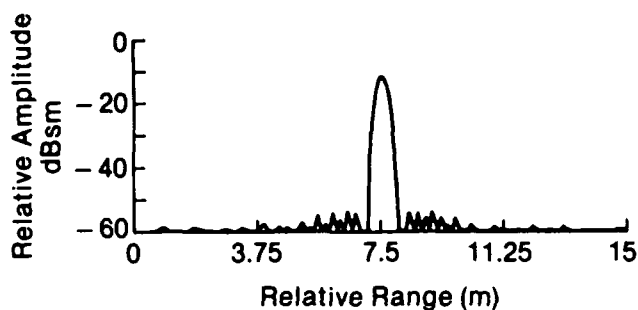


Figure 10. Effect of 30-dB system signal-to-noise ratio on 128-point FFT single-point reflector range image.

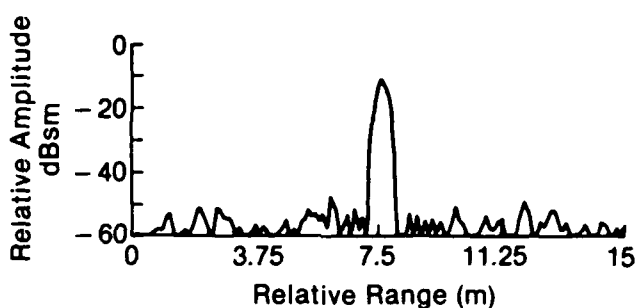


Figure 11. Effect of 20-dB system signal-to-noise ratio on 128-point FFT single-point reflector range image.

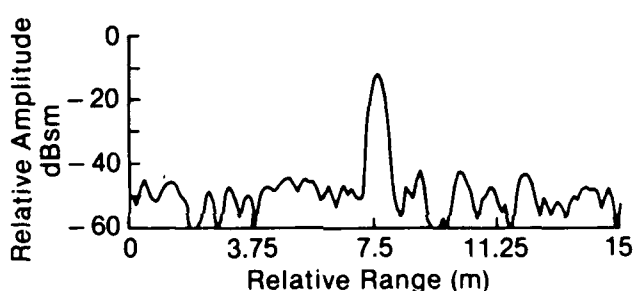
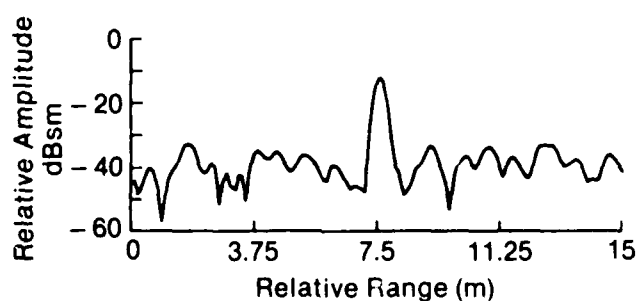


Figure 12. Effect of 10-dB system signal-to-noise ratio on 128-point FFT single-point reflector range image.



system. The first two types of phase error were modeled by adding a random phase to the phase calculated using the P_r defined by equation (8). This phase error is common to all the data channels, so both the sum and difference channels are affected equally. Simulations were run with random phase errors of 5, 10, and 20°; the results are shown in figures 13, 14, and 15. The resulting down-range images are very similar to those generated by system noise. In fact, the phase-error plot for a 5° phase error is almost identical to the system noise plot for a 20-dB signal-to-noise ratio.

The phase noise caused by errors in the I, Q detection system was not modeled separately because its effect is identical to that caused by the system noise. However, a simulation was run using a combination of the AM noise, system noise, and random phase noise. The components used for this simulation were 20 dB for the AM noise contribution, 20 dB for the system noise contribution, and 5° for the phase noise contribution. The results of this simulation are shown in figure 16. The combination of all the noise sources results in a somewhat higher background noise level than any of the individual cases.

Table 2 shows the results of averaging the background noise levels for all the noise simulations. The peak signal level for each simulation was -11.35 dBsm, so that the resulting signal-to-noise level can be calculated by dividing by the average noise levels. The results of this calculation are also shown in table 2. The signal-to-noise ratio improvement factor is about 18 dB for all the cases shown in table 2. For the combination case, the three different types of noise add up to produce three times as much noise as the individual components. This is demonstrated by an increase of the average noise for this case of approximately 5 dB over the individual cases.

Figure 13. Effect of 5° pulse-to-pulse phase variation on 128-point FFT single-point reflector range image.

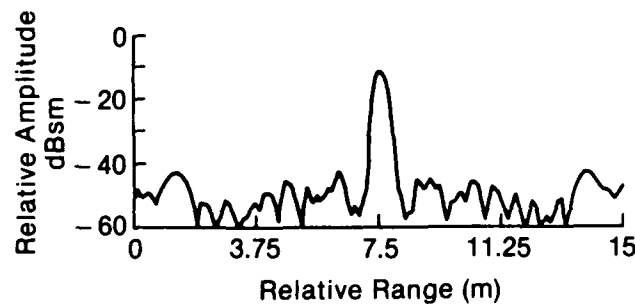


Figure 14. Effect of 10° pulse-to-pulse phase variation on 128-point FFT single-point reflector range image.

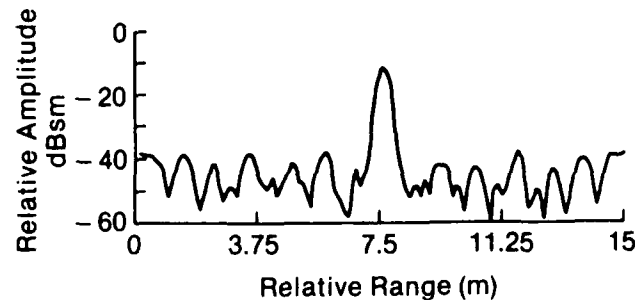


Figure 15. Effect of 20° pulse-to-pulse phase variation on 128-point FFT single-point reflector range image.

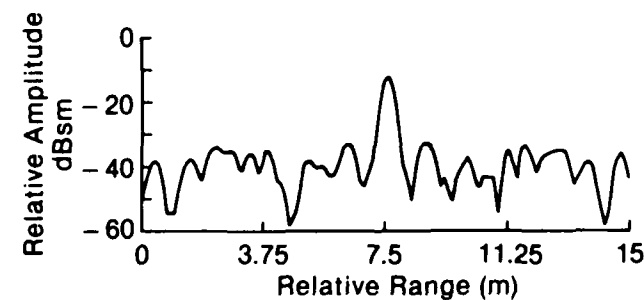


Figure 16. Effect of a combination of 20-dB system signal-to-noise ratio, -20-dB pulse-to-pulse amplitude noise, and 5° pulse-to-pulse phase variation on 128-point FFT single-point range image.

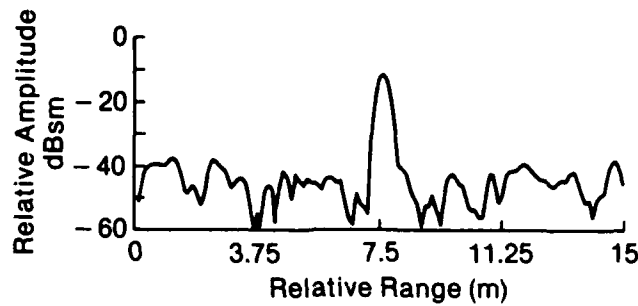


Table 2. Effect of Different Types of Noise on High-Resolution-Range FFT Imaging

Noise type	Noise amplitude (relative to signal)	Output noise (Avg. dBsm)	Output S/N (dB)
System	-40 dB	-70.6	59
	-30 dB	-58.4	47
	-20 dB	-49.3	38
	-10 dB	-40.8	29
Pulse-to-pulse	-30 dB	-60.5	49
	-20 dB	-50.7	39
	-10 dB	-39.8	28
Phase	5°	-50.2	39
	10°	-44.4	33
	20°	-38.8	27
Combination	-20dB, -20 dB, 5°	-44.1	33

4.2 Monopulse Results

An array of five isotropic point reflectors was used for the monopulse simulations. The reflectors, each with a radar cross section of 1 m², were located at the following down-range and cross-range positions: point No. 1 at 92.93 m and 0.0 m; point No. 2 at 94.93 m and +1.5 m; point No. 3 at 97.43 m and +2.0 m; point No. 4 at 97.43 m and -2.0 m; and point No. 5 at 101.93 m and -1.0 m. The positive cross-range positions indicate the left side of the azimuth beam axis, and the negative positions are for points to the right side of the azimuth axis. The elevations of the reflectors were all the same, although the simulation could have included variations of the elevation positions as well. This target array also was used in the ISAR calculations that will be discussed in the next section.

The monopulse off-axis error can be calculated for each transmit pulse or by integrating many pulses. For a long-range monopulse tracking radar, either of these methods can be used to calculate the off-axis error. For the application discussed here, the high-resolution imaging capability of a frequency-stepped radar can be used to further resolve the target in cross-range as well as in down-range, but only if there is only one target in each range bin.

Monopulse returns were calculated with system signal-to-noise ratios of 40, 30, and 20 dB. The results of processing these simulations using the FFT algorithms described earlier are shown in figures 17, 18, and 19. The top parts of the figures show the down-range image generated by FFT processing the sum pattern response of the radar. This signal was processed in the same manner as that of figure 5. The lower parts of the figures show the cross-range response generated by processing the difference channel signal in the following manner. First, the difference channel signal was processed using the FFT to provide a high-resolution down-range difference channel response. The sum channel signal was thresholded at -40 dBsm to determine the location of the different scatterers. The difference channel signal was then compared to the sum channel signal point by point to determine the normalized off-axis signal level. The calibration shown in figure 4 was used to compute the off-axis position of the target in milliradians.

The calculated positions of the targets are in excellent agreement with the input positions. The target that was positioned on-axis does not show up in the off-axis response, and the pair of points positioned symmetrically about the axis also appear to be on the azimuth axis. The sum channel response indicates a reduction in the relative amplitudes of some of the targets as compared to the input amplitudes. This effect is caused by the beam shape function employed in the simulation, which reduces the response for off-axis targets.

Figure 17. Effect of 40-dB system signal-to-noise ratio on determining off-axis target position using monopulse calibration on high-range-resolution target image data.

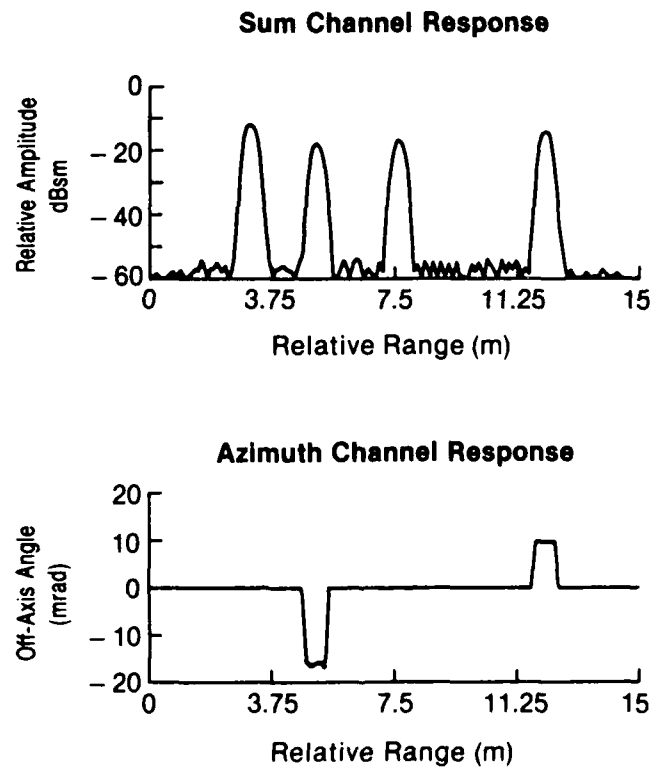


Figure 18. Effect of 30-dB system signal-to-noise ratio on determining off-axis target position using monopulse calibration on high-range-resolution target image data.

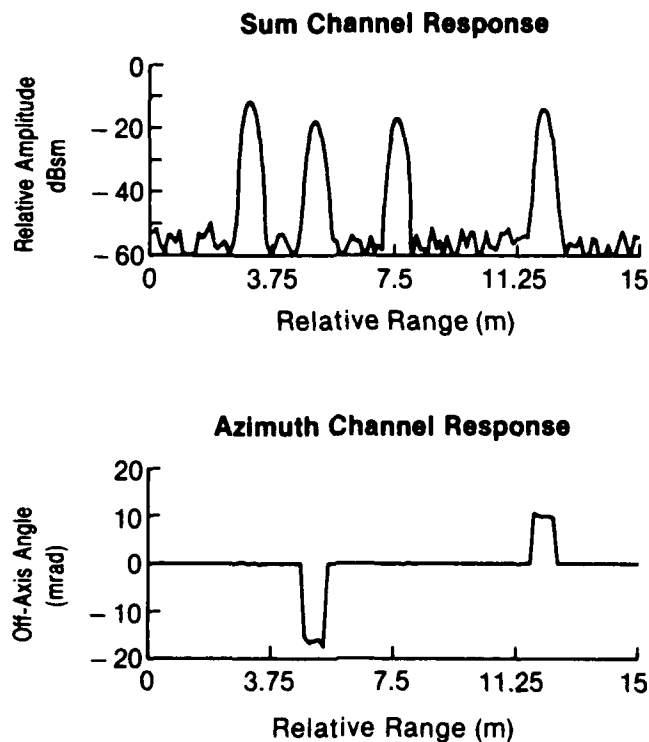
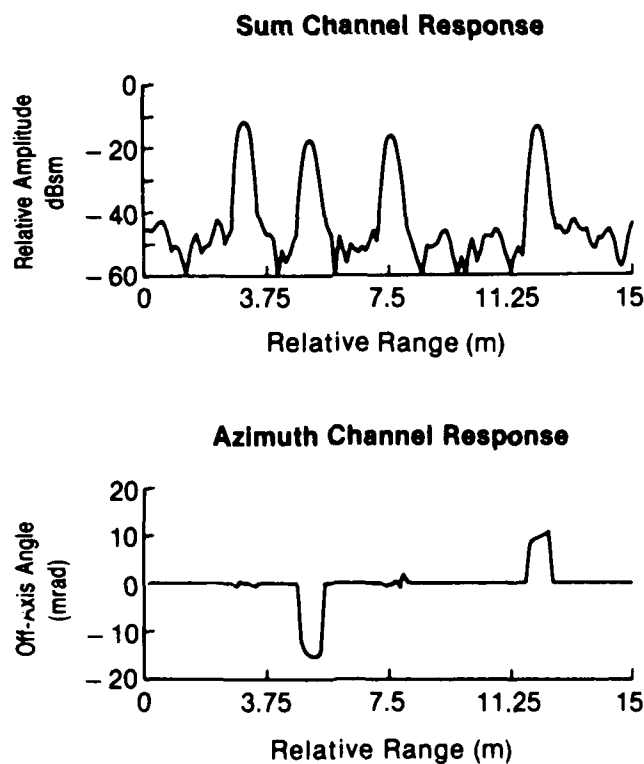


Figure 19. Effect of 20-dB system signal-to-noise ratio on determining off-axis target position using monopulse calibration on high-range-resolution target image data.



The addition of system noise has the same effect on the sum channel noise level as in the previous system noise simulations. For the off-axis response, the increase in noise is most noticeable for the targets located on-axis. Because on-axis targets have a very small difference channel signal level, the effective signal-to-noise ratio in the difference channel is very low compared to the sum channel. This effect is apparent if one examines the baseline variations of the off-axis signal for the on-axis targets as the system noise level is increased. At a system signal-to-noise ratio of 40 dB, the on-axis targets produce an almost imperceptible off-axis error signal that becomes more and more detectable as the signal-to-noise ratio is decreased to 20 dB.

This monopulse processing technique can be quite useful in resolving extended targets, because certain parts of the target can be tracked independently.

4.3 ISAR Results

The ISAR simulations were done using the same target array as used in the monopulse simulations discussed in section 4.2. The signal returned from the rotating target array was calculated as discussed in section 3.3. As in the monopulse calculations, these simulations were performed using system noise only. Because the other noise sources are special cases of system noise, it was not necessary to repeat all the calculations for either the monopulse or the ISAR simulations. Separate simulations were run using single-pulse signal-to-noise ratios of 40, 30, 20, and 10 dB. In addition a baseline simulation was run with no additive noise.

Figure 20 shows the baseline ISAR simulation calculated as discussed in section 2.2. The figure shows a hidden-line three-dimensional ISAR plot of the input target array. The range extent goes from 90 to 105 m, while the cross-range image covers 10.24 m. The amplitude scale goes from 0 dBsm to -50 dBsm, with a total dynamic range of 50 dB. The dynamic range was limited by the inability of the hidden-line program to handle multiple overlapping lines. The target array was rotated clockwise so that the right half of the plot corresponds to positive Doppler frequencies while the left half is for negative Doppler frequencies. The points that are off-axis are reduced in amplitude due to the beam shape function used to generate the raw data. With the dynamic range limited to 50 dB, the processing sidelobes, which are about 42 dB below the peak for a point target, are only visible for the two points closest to the beam axis.

The positions and amplitudes for the peak pixels for each point are given in table 3. The down-range and cross-range positions of the targets are in good agreement with the input positions, and the target amplitudes indicate the reduction due to the beam shape function.

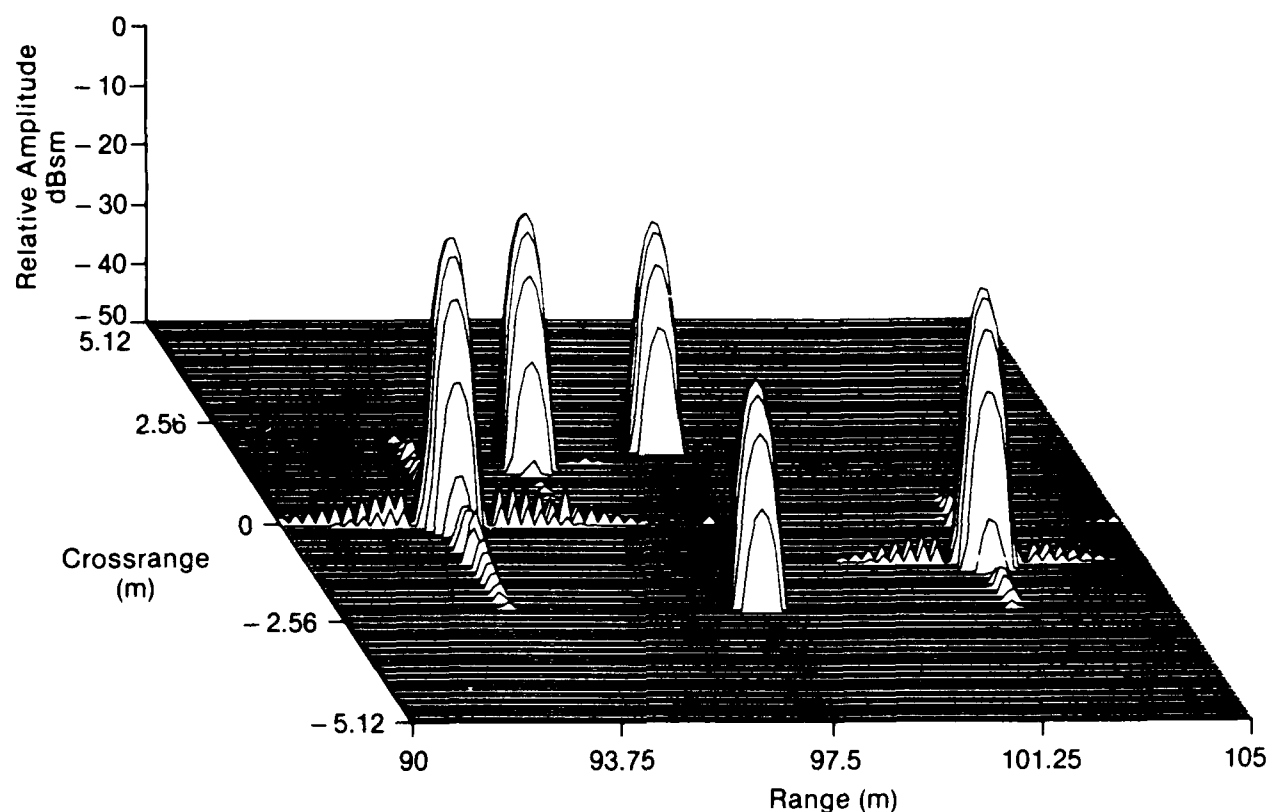


Figure 20. Three-dimensional ISAR image of target array for noiseless case.

Table 3. Comparison of Input Target Location with the Amplitude and Position Derived from ISAR Image

Input position		Output position		Amplitude (dBsm)
Range (m)	Crossrange (m)	Range (m)	Crossrange (m)	
92.93	0.0	93.03	0.0	-0.4
101.93	-1.0	102.03	-1.0	-2.7
94.93	+1.5	95.03	+1.5	-6.5
97.43	-2.0	97.43	-2.0	-11.3
97.43	+2.0	97.43	+2.0	-11.4

The effect of system noise levels on the ISAR image quality was evaluated by comparing the background noise generated in the ISAR image for the four different noise levels. The background noise was averaged over an area 32 pixels on a side. This area, covering the upper right-hand corner of figure 20, was selected so that processing sidelobes generated by each target were excluded as much as possible.

The results are shown in table 4. The table gives the input single-pulse signal-to-noise ratio, the average noise over the 32×32 pixel area, and the calculated output signal-to-noise ratio for each case. The table indicates an improvement

in the output signal-to-noise ratio of approximately 34 dB. The theoretical improvement in signal-to-noise ratio for coherently integrating 4096 separate samples of signal plus noise is 36 dB. The results are in good agreement with this calculation; the slight difference is probably due to the addition of a small amount of sidelobe power from the individual points.

The ISAR image for a 20-dB input signal-to-noise ratio is shown in figure 21. The noise is apparent, but it would not greatly limit the ability to resolve targets of different amplitudes in the image. Targets as much as 30 dB below the peak target amplitude could still be distinguished from the background noise. Because the sidelobe level limits the dynamic range to about 40 dB for scattering points physically close together, an input single-pulse signal-to-noise ratio of 20 dB is probably adequate for this kind of image processing.

Table 4. Comparison of Average Output Noise Level for ISAR Processing with Input Noise Level

Input S/N ratio (dB)	Average noise (dBsm)	Output S/N ratio (dB)
∞	-77.3	77
40	-71.1	71
30	-64.0	64
20	-54.8	55
10	-44.1	44

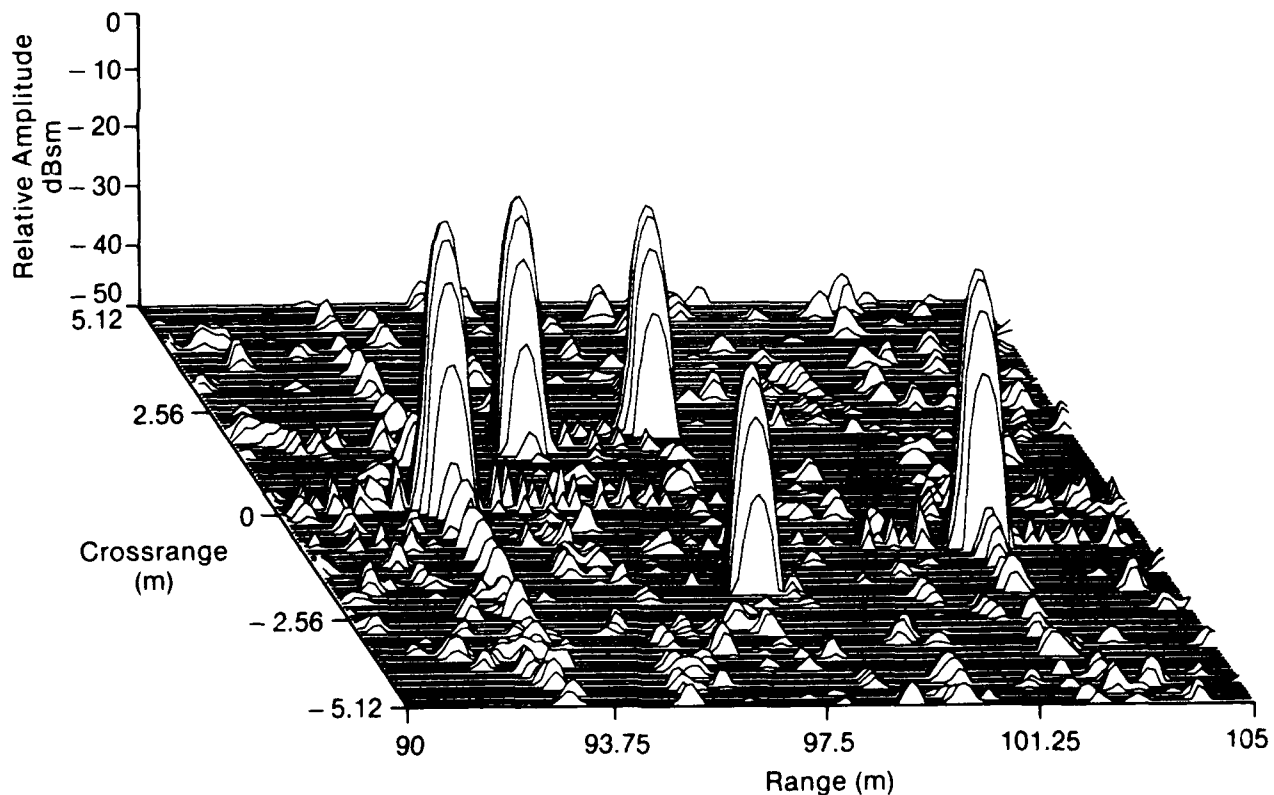


Figure 21. Three-dimensional ISAR image of target array for 20-dB input signal-to-noise ratio.

5. Conclusions

We have developed a radar simulation program with the capability of generating frequency-stepped coherent monopulse signature data from input target arrays. The program allows one to calculate the return from a three-dimensional target array for different elevation and azimuth angles. The program has been demonstrated for a target array of point reflectors, but it could be used to calculate target signatures from target model data. In addition, system parameters such as antenna size, beam squint angle, radar frequency, frequency step size, and the number of frequency steps per ramp can be changed at will. By varying these parameters, different radar systems can be simulated.

We have applied FFT processing techniques to the simulated frequency-stepped data to generate high-range-resolution images of target arrays. These techniques use padding to minimize range response variations and windowing to improve sidelobe performance. In addition, various noise sources have been added to the simulated data to determine their effects on range images produced with FFT processing. For our application, the 18-dB processing gain associated with this technique makes it possible to generate reasonable image quality with single-pulse input signal-to-noise ratios as low as 10 dB. For applications where a number of individual scatterers with a wide range of amplitudes need to be resolved, a slightly higher input signal-to-noise ratio would be required. We have also applied these high-range resolution methods to the monopulse azimuth channel signal to allow the calculation of the off-axis position of scatterers within the target array. This technique could be applied to monopulse data collected while tracking a target, to select a particular part of the target to track. Alternatively, this method could be used to separate the target sum and difference signals in range from the background signals. This would allow more accurate tracking of a target in a high clutter environment.

Using the target simulation program, we have generated rotating target signature data and calculated two-dimensional ISAR images of the target array. We have developed an ISAR image-processing program to analyze these data and any other coherent frequency-agile turntable data that are available. This program uses sets of frequency-stepped data at different aspect angles to generate the images. Since the input array consists of 4096 different looks at the target, there is a 36-dB improvement in the signal-to-noise ratio after processing. This means that even at single-pulse signal-to-noise ratios of 20 dB, the image quality is not significantly degraded. This was demonstrated by generating ISAR images of a target array with an input signal-to-noise ratio of 20 dB. The generated image accurately reproduced both the amplitude and position of the various scatterers in the input array.

We have developed and demonstrated a set of programs for generating and processing coherent frequency-agile radar data for use primarily in target signature analysis. These programs will be used to analyze target signature data that will be collected with the Harry Diamond Laboratories frequency-agile fully polarimetric coherent monopulse 95-GHz radar.

References

1. Merrill I. Skolnik, *Introduction to Radar Systems*, McGraw-Hill Book Company, New York (1980).
2. Robert W. Ramirez, *The FFT Fundamentals and Concepts*, Prentice-Hall Inc., New Jersey (1985).
3. Dean L. Mensa, *High Resolution Radar Imaging*, Artech House, Inc., Massachusetts (1984).
4. August W. Rihaczek, *Principles of High-Resolution Radar*, McGraw-Hill Book Company, New York (1969).
5. D. C. Munson, Jr., J. D. O'Brien, and W. K. Jenkins, *A Tomographic Formulation of Spotlight-Mode Synthetic Aperture Radar*, Proceedings of the IEEE, **71**, 8 (August 1983), 917-925.
6. C. C. Chen and H. C. Andrews, *Multifrequency Imaging Radar Turntable Data*, IEEE Transactions on Aerospace and Electronic Systems, **AES-16**, 1 (January 1980), 15-22.

**Appendix A. Listing of stepped-frequency monopulse
signal calculation program**

Appendix A

```

      program isargen64_plus
c*****
c  This program computes phases and amplitudes from
c  point sources. The output of the program is saved
c  to a disk file.
c      Version 1.0 (MMW) written for HP-1000 3/86
c      Version 2.0 (siggen) written for C-1 3/87
c      Version 2.1 written for 64x64 arrays 4/87
c      Version 2.2 depression and azimuth angles 4/87
c      Version 2.5 added noise,output format simplified
c      Version 3.0 Noise delegated to freq-loop 4/88
c      Version 3.1 Inter-ramp dead-time added 4/88
c      Written by Jerry Silvious , Branch 11300 HDL
c
c  Known Bugs: Does not handle depression <> 0 properly
c*****
      implicit none

      integer*4 j
      integer*4 stdout                !standard output unit

      stdout=6

      call isargen

      write(stdout,*)
      write(stdout,(' ** ISAR Simulation Terminating **'))
      end

c*****
c      ISAR TARGET SIGNAL GENERATOR
c
c  This routine computes phases and amplitudes of point
c  sources for various frequencies and various observation
c  angles. The locations of the points is read from a disk
c  file. The output is stored to another disk file.
c*****
      subroutine isargen

      implicit none

      complex*8 cangle                !locates angle of points

      integer*2 ntype(4)              !noise type

      integer*4 h,k,l,m               !counter variables
      integer*4 stdin                 !standard in
      integer*4 stdout                 !standard out
      integer*4 stderr                 !standard errfile
      integer*4 uniti                 !input unit for data
      integer*4 unito                 !output unit for data
      integer*4 pts                   !number of point sources
      integer*4 channel               !beam number counter
      integer*4 iargc,args            !command line arguments
      integer*4 stime                 !opsys time
      integer*4 irand                 !integer random number function

```

```

real*4 d0(4096),r0(4096),h0(4096)  !point location buffers
real*4 phi0                        !initial aspect angle
real*4 phi                          !updated aspect angle
real*4 circlrad                     !radius from centroid
real*4 depression                   !depression angle
real*4 rho                          !distance to center of rotation
real*4 azimuth                      !azimuth angle of antenna
real*4 ax                           !crossrange coordinate of antenna
real*4 ay                           !downrange
real*4 az                           !height of antenna
real*4 ramp_cycles                  !Number of ramps to be executed
real*4 time                         !time variable
integer*4 null_ramps                !ramps for which there is no data
real*4 steps                        !frequency step size
integer*4 nsteps                    !number of steps
real*4 deg_per_ramp                 !rotation rate
real*4 convert                      ! pi/180
real*4 norm

real*4 nampr                        !noisy amplitude component
real*4 vec_phase(3)                 !vector phase
real*4 vec_amp(3)                   !vector amplitude
real*4 beam_peak
real*4 new_phase                    !noisy total phase component
real*4 amp(4096)                    !point amplitude array
real*4 ajit(2),pjit(2)              !noise factors
real*8 dbuff(6)                     !real-valued databuffer

real*8 bfreq                        !beam parameters
real*8 warp                         !speed of light (m/sec)
real*8 freq                         !updated frequency
real*8 lambda                       !wavelength

real*8 pi
real*8 theta                        !computed phase angle
real*8 hypotenuse                   !radial distance for phase calculation
real*8 new_d0                       !updated downrange location
real*8 a0,radius,height             !updated point locations
real*8 squint
real*8 dia
real*8 two

real*4 rnd(6),distribution(4096)    !random number collection
real*4 rand                         !opsys random number function
real*4 unique_amp(3)                !system noise component
real*4 unique_phase(3)              !random system noise phase
real*4 outbuff(3072)                !output buffer

character infile*80,outfile*80

parameter ( stdin = 5 )
parameter ( stdout= 6 )
parameter ( stderr = 0 )
parameter ( uniti =10 )
parameter ( unito = 20 )

```


Appendix A

```

parameter ( warp = 299792458. )
parameter ( convert = .017453292 )      ! pi/180.
parameter ( pi = 3.14159265358979 )

parameter ( bfreq = 94.44E9 )           !transmit frequency 95 GHz
parameter ( squint=7.443745e-3 )        ! .3125*.02382 (see below)
parameter ( dia=.15748 )                !6.2" antenna diameter
parameter ( dia=.0381 )                 !1.5" antenna diameter
parameter ( null_ramps = 8 )            ! no data during 2 ramps
common /specs/hypotenuse,ax,ay,az,a0,radius,height

two = dble(2.)

c***** Recover command line arguments

args=iargc()
if (args.lt.2) then
  write(stderr,*) 'Usage:      isargen64+ input_file output_file'
  stop
endif

write(stderr,*)
write(stderr,*)('***** ISAR SIGNAL GENERATION *****')

call getarg(1,infile)
open(unit1,file=infile,status='old',READONLY)

call getarg(2,outfile)
open(unit0,file=outfile,STATUS='unknown')

write(stderr,*)
write(stderr,*)('Radial distance to target (m) | ",$')
read(unit1,*) rho
write(stderr,*) rho
write(stderr,*)('Depression angle (deg) | ",$')
read(unit1,*) depression
write(stderr,*) depression
write(stderr,*)('Azimuth angle (deg) | ",$')
read(unit1,*) azimuth
write(stderr,*) azimuth
write(stderr,*)('Rotation speed (deg/ramp-ramp) | ",$')
read(unit1,*) deg_per_ramp
write(stderr,*) deg_per_ramp

write(stderr,*)('Ramp cycles | ",$')
read(unit1,*) ramp_cycles
write(stderr,*) ramp_cycles
write(stderr,*)('Frequency step size (MHz) | ",$')
read(unit1,*) steps
write(stderr,*) steps
write(stderr,*)('Number of steps per ramp | ",$')
read(unit1,*) nsteps
write(stderr,*) nsteps
write(stderr,*)
write(stderr,*)('*****')

```

```

write(stderr, '("Type return to continue...") ')
read(stdin, *)

read(uniti, *) pts                !read the number of points

do k=1,pts                        !read in the point coordinates
    read(uniti, *) d0(k), r0(k), h0(k), amp(k)
enddo

read(uniti, *) ntype(1), ntype(2), ntype(3), ntype(4)
read(uniti, *) ajit(1), pjit(1), ajit(2), pjit(2)        !jitter values

close(uniti)

open(10, file='random_nos', READONLY)        !load random distribution
do k=1, 4096                                !array
    read(10, *) distribution(k)
enddo
close(10)
c ***** begin data generation *****
time=-1.*((deg_per_ramp)/nsteps/(null_ramps+1))        !set time increment
                                                    !antenna location
c    ax=rho*cos(depression*pi/180.)*sin(azimuth*pi/180.) !X-range
c    ay=rho*cos(depression*pi/180.)*cos(azimuth*pi/180.) !D-range
c    az=rho*sin(depression*pi/180.)                    !height
ax = 0
ay = rho
az = 0

do m=1, ramp_cycles                    !number of angles loop
    write(stderr, ' (a16,i4,a1,$) ') 'Computing Ramp #', m, char(13)

    time=(m-1)*deg_per_ramp                !rotation w/ no-data
    do l=1, nsteps                        !number of frequency pulses loop
        freq=bfreq+float(l-1)*steps*1.E6    !step the frequency
        lambda=warp/freq                    !compute the wavelength
        do h=1, 6
            dbuff(h)=0.                    !clear data buffer
        enddo

        time=time+((deg_per_ramp)/nsteps/(null_ramps+1))
c        phi=+1.*time*pi/180.                ! CW rotation \aspect angle
c        phi=-1.*time*pi/180.                !CCW rotation / (?)

        do k=1, pts                        !# of points loop
c ***** compute the return phase
            new_d0=d0(k)
c            new_d0=d0(k)+(.234212857)*(1-1)/(160.*nsteps)    !walking point

            circlrad=sqrt(new_d0*new_d0+r0(k)*r0(k))

            cangle=cmplx(dble(r0(k)), new_d0)    !compute the initial angle
            call kwik(cangle, phi0)
            if (circlrad.eq.0.) phi0=0.            !point at center of rot.
            phi0 = phi0 + azimuth*pi/180.        ! user rotational offset

```

Appendix A

```

a0=circlrad*dsin(dble(phi-phi0))           !update D-range coord.
radius=circlrad*dcos(dble(phi-phi0))       !update V-range coord.
height=h0(k)                               !(cylindrical trajectory)

hypotenuse=(ay-a0)*(ay-a0)+(ax-radius)*(ax-radius)
&      +(az-height)*(az-height)

theta=dsqrt(hypotenuse)*(two*(two*pi/lambda))
theta=dmod(theta,(two*pi)) !phase,mod 2*pi

call nominal(freq,amp(k),theta,dbuff) !compute the amplitudes
enddo

beam_peak=norm(freq,squint,dia)            !compute noise-ref factor

cangle=cplx(sngl(dbuff(1)),sngl(dbuff(2)))
call kwik(cangle,vec_phase(1))
vec_amp(1)=sngl(dsqrt(dbuff(1)*dbuff(1)+dbuff(2)*dbuff(2)))
cangle=cplx(sngl(dbuff(3)),sngl(dbuff(4)))
call kwik(cangle,vec_phase(2))
vec_amp(2)=sngl(dsqrt(dbuff(3)*dbuff(3)+dbuff(4)*dbuff(4)))

cangle=cplx(sngl(dbuff(5)),sngl(dbuff(6)))
call kwik(cangle,vec_phase(3))
vec_amp(3)=sngl(dsqrt(dbuff(5)*dbuff(5)+dbuff(6)*dbuff(6)))

c ***** compute noise factors *****
rnd(1) = m*float(irand(stime()))/1.E6+float(1-1)
rnd(2) = m*float(irand(stime()))/1.E6+64.+float(1-1)
rnd(3) = m*float(irand(stime()))/1.E6+128.+float(1-1)
rnd(4) = m*float(irand(stime()))/1.E6+192.+float(1-1)
rnd(5) = m*float(irand(stime()))/1.E6+256.+float(1-1)
rnd(6) = m*float(irand(stime()))/1.E6+320.+float(1-1)
rnd(1)=distribution(int(4096.*rand(1.E6*int(rnd(1)))+.5))
rnd(2)=distribution(int(4096.*rand(1.E6*int(rnd(2)))+.5))
rnd(3)=distribution(int(4096.*rand(1.E6*int(rnd(3)))+.5))
rnd(4)=rand(1.E6*int(rnd(4)))
rnd(5)=rand(1.E6*int(rnd(5)))
rnd(6)=rand(1.E6*int(rnd(6)))

if (ntype(1).eq.1) then                    !pulse-to-pulse amplitude
  nampr=rnd(1)*10**(-ajit(1)/20.)
else
  nampr=0.
endif

if (ntype(2).eq.1) then                    !pulse-to-pulse phase
  new_phase=dble(pjit(1)*rnd(1)*convert)
else
  new_phase=0.
endif

if (ntype(3).eq.1) then
  unique_amp(1)=rnd(1)*10**(-ajit(2)/20.)
  unique_amp(2)=rnd(2)*10**(-ajit(2)/20.)

```

```

        unique_amp(3)=rnd(3)*10**(-ajit(2)/20.)
        unique_phase(1) = rnd(4)*2.*pi
        unique_phase(2) = rnd(5)*2.*pi
        unique_phase(3) = rnd(6)*2.*pi
    else
        unique_amp(1)=0.
        unique_amp(2)=0.
        unique_amp(3)=0.
        unique_phase(1)=0.
        unique_phase(2)=0.
        unique_phase(3)=0.
    endif
c*****
    dbuff(1)=(vec_amp(1)+nampr)*cos(vec_phase(1)+new_phase)
&          +unique_amp(1)*cos(unique_phase(1))
    dbuff(2)=(vec_amp(1)+nampr)*sin(vec_phase(1)+new_phase)
&          +unique_amp(1)*sin(unique_phase(1))
    dbuff(3)=dbuff(3)
&          +unique_amp(2)*cos(unique_phase(2))
    dbuff(4)=dbuff(4)
&          +unique_amp(2)*sin(unique_phase(2))
    dbuff(5)=dbuff(5)
&          +unique_amp(3)*cos(unique_phase(3))
    dbuff(6)=dbuff(6)
&          +unique_amp(3)*sin(unique_phase(3))

    outbuff((1-1)*6+1)=sngl(dbuff(1))      !i sum
    outbuff((1-1)*6+2)=sngl(dbuff(2))      !q sum
    outbuff((1-1)*6+3)=sngl(dbuff(3))      !i az
    outbuff((1-1)*6+4)=sngl(dbuff(4))      !q az
    outbuff((1-1)*6+5)=sngl(dbuff(5))      !i el
    outbuff((1-1)*6+6)=sngl(dbuff(6))      !q el

    enddo                                     !end of frequency loop
    write(unito,*) (outbuff(k),k=1,6*nsteps)
enddo                                       !end of aspect angle

close(unito)                               !close output file

return
end
c*****
subroutine nominal(freq,mag,theta,dbuff)
implicit none

integer*4 channel
integer*4 table(4,4)

real*4 mag
real*4 ampr
real*4 champ
real*4 norm
real*4 beam_peak

real*8 dbuff(6)

```

Appendix A

```

      real*8 squint
      real*8 dia
      real*8 in,q
      real*8 freq
      real*8 theta

      data table/1,0,0,0,
&          0,1,0,0,
&          0,0,1,0,
&          0,0,0,1/

      parameter ( squint=7.443745e-3 )      ! .3125*.02382 (see below)
c      parameter ( dia=.15748 )             !6.2" antenna diameter
      parameter ( dia=.0381 )

      beam_peak=norm(freq,squint,dia)      !compute sum-norm factor
c***** SUMMATION CHANNELS *****
      do channel=1,4

          ampr=(10**(mag/20.))*champ(freq,channel)
          in=dbl(ampr)*dcos(theta)/beam_peak      !in-phase
          q =dbl(ampr)*dsin(theta)/beam_peak      ! q-phase

          dbuff(1)=dbuff(1)+in                    !i sum channel
          dbuff(2)=dbuff(2)+q                      !q sum channel
      enddo

c***** DIFFERENCE CHANNELS *****
      do channel=1,4
          ampr=(10**(mag/20.))*champ(freq,channel)

          in=dbl(ampr)*dcos(theta)/beam_peak      !in-phase signal
          q =dbl(ampr)*dsin(theta)/beam_peak      ! q-phase signal

c azimuth i:
          dbuff(3)=dbuff(3)+in*table(1,channel)+in*table(3,channel)
          &          -in*table(2,channel)-in*table(4,channel)
c azimuth q:
          dbuff(4)=dbuff(4)+q*table(1,channel)+q*table(3,channel)
          &          -q*table(2,channel)-q*table(4,channel)
c elevation i:
          dbuff(5)=dbuff(5)+in*table(1,channel)+in*table(2,channel)
          &          -in*table(3,channel)-in*table(4,channel)
c elevation q:
          dbuff(6)=dbuff(6)+q*table(1,channel)+q*table(2,channel)
          &          -q*table(3,channel)-q*table(4,channel)
      enddo

      return
      end

```

Appendix A

```

c*****
c      Channel amplitude function      *
c*****
      real function champ(freq,channel)
      implicit none

      real*4 ajit                !amplitude jitter
      real*4 norm
      real*4 ax                  !crossrange coordinate of antenna
      real*4 ay                  !downrange      '      '
      real*4 az                  !height of antenna

      real*8 hypotenuse          !absolute range to point
      real*8 a0                  !relative downrange location
      real*8 radius              !crossrange location
      real*8 height              !height of point source

      real*8 freq                !transmitted frequency
      real*8 squint              !half-angle between beams
      real*8 dia                 !antenna diameter
      real*8 c                   !speed of light
      real*8 psi                 !angle for amplitude equation
      real*8 length              !denominator of dot product
      real*8 gamma               !dot product angle
      real*8 by,bz,bx            !nearest point on beam
      real*8 amp                 !amplitude (not normalized)

      integer*4 channel          !beam channel

      parameter ( squint=7.443745e-3 )      ! .3125*.02382 half-angle between
                                          ! beam centers
c      parameter ( dia=.15748 )            !6.2" antenna diameter
      parameter ( dia=.0381 )              !6.2" antenna diameter
      parameter ( pi=3.14159265358979 )
      parameter ( c=299792458. )

      common /specs/hypotenuse,ax,ay,az,a0,radius,height

      by=dsqrt(hypotenuse)                !use range calculation
      bz=abs(by*dtan(squint))              !height=off-axis distance
      if (channel.eq.1) bx=bz              !determine closest point
      if (channel.eq.2) bx=-bz             ! on the given beam
      if (channel.eq.3) then
         bx=bz
         bz=-bz
      endif
      if (channel.eq.4) then
         bz=-bz
         bx=bz
      endif

      length=dsqrt(bx*bx+by*by+bz*bz)*dsqrt(hypotenuse) !denominator .product

```

Appendix A

```

    if (length.eq.0) then
        gamma=0.
    else
        gamma=dacos((bx*(ax-radius)+by*(ay-a0)+bz*(az-height))
&                /length)!dot product angle
    endif

    psi=pi*dia*dsin(gamma)/(c/freq)

    amp=dsin(psi+pi/2.)/(psi+pi/2.)    !amplitude
&+dsin(psi-pi/2.)/(psi-pi/2.)
    champ=(amp*pi/4.)*(amp*pi/4.)    !normalized beam amplitude

    return
end

c ***** Normalization Factor *****
real function norm(freq,squint,dia)
implicit none

real*8 freq                !transmitted frequency
real*8 squint              !beam squint angle
real*8 dia
real*8 bz
real*8 gamma,psi
real*8 c
real*8 pi
real*8 amp
real*8 len

parameter ( pi = 3.14159265358979 )
parameter ( c = 299792458 )

bz=1000.*dtan(dble(squint))
len=1000.*sqrt(bz*bz+1000.*1000.+bz*bz)
gamma=dacos((1000.*1000.)/len)
psi=pi*dia*sngl(dsin(gamma))/(c/freq)
amp=dsin(psi+pi/2.)/(psi+pi/2.)+dsin(psi-pi/2.)/(psi-pi/2.)
norm=4.*(amp*pi/4.)*(amp*pi/4.)

return
end

c *****
c                               KWIK                               *
c    This routine computes the complex                             *
c    angle of a given complex number.                             *
c *****
subroutine kwik(cangle,phi0)

complex*8 cangle                !input complex number
real*4 phi0                    !output angle
real*4 pi
real*4 r0                      !real part of cangle
real*4 d0                      !imaginary part of cangle

```

```
pi=3.14159265358979

r0=real(cangle)
d0=aimag(cangle)

if ((r0.eq.0.).and.(d0.ge.0.)) then
    phi0=pi/2.
    goto 100
endif
if ((r0.eq.0.).and.(d0.lt.0.)) then
    phi0=3*pi/2.
    goto 100
endif

phi0=atan(d0/r0)

if ((r0.gt.0.).and.(d0.ge.0.)) phi0=phi0
if ((r0.lt.0.).and.(d0.ge.0.)) phi0=phi0+pi
if ((r0.lt.0.).and.(d0.lt.0.)) phi0=phi0+pi
if ((r0.gt.0.).and.(d0.lt.0.)) phi0=phi0+2*pi

100  return
end
```


DISTRIBUTION

ADMINISTRATOR
DEFENSE TECHNICAL INFORMATION
CENTER
ATTN DTIC-DDA (2 COPIES)
CAMERON STATION, BUILDING 5
ALEXANDRIA, VA 22304-6145

DIRECTOR
US ARMY BALLISTIC RESEARCH
LABORATORY
ATTN DIRECTOR, SLCBR-D
ATTN SLCBR-SE-W, H. B. WALLIC
ABERDEEN PROVING GROUND, MD 21005

US ARMY ELECTRONIC TECHNOLOGY
& DEVICES LABORATORY
ATTN SLCET-DD
FT MONMOUTH, NJ 07703

DIRECTOR
US ARMY MATERIEL SYSTEMS
ANALYSIS ACTIVITY
ATTN AMXSY-MP
ABERDEEN PROVING GROUND, MD 21005

DEPT OF THE AIR FORCE, HQ
6585TH TEST GROUP (AFSC)
RADAR TARGET SCATTER FACILITY
ATTN LT COL R. L. KERCHER, CHIEF
HOLLOMAN AFB, NM 88330

ENGINEERING SOCIETIES LIBRARY
ATTN ACQUISITIONS DEPARTMENT
345 EAST 48TH STREET
NEW YORK, NY 10017

NATIONAL OCEANIC & ATMOSPHERIC ADM
ENVIRONMENTAL RESEARCH LABORATORIES
ATTN LIBRARY, R-51, TECH REPORTS
BOULDER, CO 80302

DIRECTOR
DEFENSE ADVANCED RESEARCH
PROJECTS AGENCY
ATTN TARGET ACQUISITION
& ENGAGEMENT DIV
1400 WILSON BLVD
ARLINGTON, VA 22209

DIRECTOR
NATIONAL SECURITY AGENCY
ATTN TECHNICAL LIBRARY
FT MEADE, MD 20755

UNDER SECRETARY OF DEFENSE
FOR RESEARCH & ENGINEERING
ATTN RESEARCH & ADVANCED TECH
WASHINGTON, DC 20301

DIRECTOR
APPLIED TECHNOLOGY LABORATORY
AVRADCOM
ATTN DAVDL-ATL-TSD, TECH LIBRARY
FT EUSTIS, VA 23604

COMMANDER
US ARMY ARMAMENT, MUNITIONS
& CHEMICAL COMMAND
ATTN DRDAR-TDR, RESEACH
& TECHNOLOGY
DOVER, NJ 07801-5001

COMMANDER/DIRECTOR
ATMOSPHERIC SCIENCES LABORATORY
ATTN DELAS-AS, ATMOSPHERIC SENSING DIV
WHITE SANDS MISSILE RANGE, NM 88002

COMMANDER
US ARMY COLD REGIONS RESEARCH
& ENGINEERING LABORATORY
ATTN CRREL-TI, TECHNICAL INFO BR
HANOVER, NJ 03755

COMMANDER
CECOM
CENTER FOR EW/RSTA
ATTN AMSEL-RD-EW-R, A. TARBELL
FT MONMOUTH, NJ 07703-5300

COMMANDER
US ARMY MATERIEL COMMAND
ATTN AMCDM, DIR FOR PLANS & ANALYSIS
5002 EISENHOWER AVE
ALEXANDRIA, VA 22333-0001

DIRECTOR
US ARMY MISSILE LABORATORY
ATTN AMSMI-RPR, REDSTONE SCIENTIFIC
INFO CENTER
ATTN AMSMI-RPT, TECHNICAL INFORMATION
DIV
ATTN AMSMI-RD, SYS SIMULATION * DEV
DIR
ATTN AMSMI-RD, ADVANCED SENSORS DIR
ATTN AMSMI-RD-AS-MM, G. EMMONS
ATTN AMSMI-RD-AS-RA, J. LOOMIS
REDSTONE ARSENAL, AL 35809

DIRECTOR
VIGHT VISION & ELECTRO-OPTICS
CENTER
ATTN TECHNICAL LIBRARY
ATTN CNVEO-AC, ADVANCED CONCEPTS
DIV
FT BELVOIR, VA 22060

DISTRIBUTION (cont'd)

US CHIEF ARMY RESEARCH OFFICE
(DURHAM)
ATTN DRXRO-EG, DIR ENGINEERING DIV
PO BOX 12211
RESEARCH TRIANGLE PARK, NC 27709

COMMANDER
US ARMY TEST & EVALUATION
COMMAND
ATTN STEWS-TE-LG, S. DICKERSON
ATTN STEWS-TE-AF, F. MORENO
WHITE SANDS MISSILE RANGE, NM 88002

DIRECTOR
NAVAL RESEARCH LABORATORY
ATTN 2600, TECHNICAL INFO DIV
WASHINGTON, DC 20375

COMMANDER
NAVAL SURFACE WARFARE CENTER
ATTN DX-21 LIBRARY DIV
DAHLGREN, VA 22448

COMMANDER
NAVAL SURFACE WARFARE CENTER
ATTN E-43, TECHNICAL LIB
WHITE OAK, MD 20910

COMMANDER
NAVAL WEAPONS CENTER
ATTN 38, RESEARCH DEPT
ATTN 381, PHYSICS DIV
CHINA LAKE, CA 93555

HEADQUARTERS
ARMAMENT DIV (AFSC)
ATTN MSD/ENG, MAJ M. TURK
EGLIN AFB, FL 32542

DIRECTOR
NASA
GODDARD SPACE FLIGHT CENTER
ATTN 250, TECH INFO DIV
GREENBELT, MD 20771

DIRECTOR
NASA
LANGLEY RESEARCH CENTER
ATTN TECHNICAL LIBRARY
HAMPTON, VA 23665

ENVIRONMENTAL RESEARCH INSTITUTE
OF MICHIGAN
ATTN IRIA LIBRARY
PO BOX 618
ANN ARBOR, MI 48107

GEORGIA TECH RESEARCH INSTITUTE
GEORGIA INSTITUTE OF TECHNOLOGY
RADAR & INSTRUMENTATION LABORATORY
ATTN T. L. LANE
ATLANTA, GA 30332

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF
TECHNOLOGY
ATTN R. M. BARNES
LEXINGTON, MA 02173

SANDIA NATIONAL LABORATORIES
PO BOX 5800
ALBUQUERQUE, NM 87185

SIMULATION TECHNOLOGIES, INC
ATTN A. V. SAYLOR
PO BOX 7009
HUNTSVILLE, AL 35807

TASC
ATTN K. L. WEEKS
907 MAR-WALT DRIVE
FT WALTON BEACH, FL 32548

US ARMY LABORATORY COMMAND
ATTN TECHNICAL DIRECTOR, AMSLC-TD
ATTN DIRECTOR, AMCLD-SE (N. BERG)

INSTALLATION SUPPORT ACTIVITY
ATTN LEGAL OFFICE, SLCIS-CC

USAISC
ATTN RECORD COPY, AMSLC-IM-TS
ATTN TECHNICAL REPORTS BRANCH,
AMSLC-IM-TX (3 COPIES)

HARRY DIAMOND LABORATORIES
ATTN D/DIVISION DIRECTORS
ATTN DIRECTOR, SLCHD-D
ATTN LIBRARY, SLCHD-TL (3 COPIES)
ATTN LIBRARY, SLCHD-TL (WOODBIDGE)
ATTN CHIEF, SCIENTIST, SLCHD-CS
ATTN DEPUTY DIRECTOR, SLCHD-DD
ATTN ASSOC DIRECTOR, SLCHD-PO
ATTN DIRECTOR, SLCHD-ST
ATTN DIRECTOR, SLCHD-NW
ATTN DIRECTOR, SLCHD-TA
ATTN DIRECTOR, SLCHD-TS
ATTN CHIEF, SLCHD-ST-AR
ATTN CHIEF, SLCHD-ST-SS
ATTN CHIEF, SLCHD-ST-MW
ATTN CHIEF, SLCHD-ST-OP
ATTN CHIEF, SLCHD-ST-SP

DISTRIBUTION (cont'd)

HARRY DIAMOND LABORATORIES

(cont'd)

ATTN CHIEF, SLCHD-ST-R
ATTN CHIEF, SLCHD-ST-SA
ATTN CHIEF, SLCHD-ST-MS
ATTN CHIEF, SLCHD-ST-AP
ATTN R. BERG, SLCHD-NW-EH
ATTN S. KHAN, SLCHD-NW-EH
ATTN W. J. PATTERSON, SLCHD-NW-EH
ATTN P. WALKER, SLCHD-ST
ATTN D. R. COOK, SLCHD-ST-AR
ATTN D. L. RODKEY, SLCHD-ST-MS
ATTN J. DAVID, SLCHD-ST-R
ATTN C. ARSEM, SLCHD-ST-SA
ATTN J. COSTANZA, SLCHD-ST-SA

HARRY DIAMOND LABORATORIES

(cont'd)

ATTN D. M. HULL, SLCHD-ST-SA
ATTN H. LE, SLCHD-ST-SA
ATTN J. NEMARICH, SLCHD-ST-SP
ATTN J. DROPKIN, SLCHD-ST-SP
ATTN G. GOLDMAN, SLCHD-ST-SP
ATTN D. WIKNER, SLCHD-ST-SP
ATTN D. HUTCHINGS, SLCHD-ST-SP
ATTN M. PATTERSON, SLCHD-ST-SS
ATTN J. SILVERSTEIN, SLCHD-ST-SP
ATTN R. KAPOOR, SLCHD-TA-AS (2 COPIES)
ATTN R. D. GOODMAN, SLCHD-TA-ES
ATTN J. SILVIOUS, SLCHD-ST-SP (20 COPIES)
ATTN R. WELLMAN, SLCHD-ST-SP (20 COPIES)